Designing Survivable Networks: A Flow Based Approach

Prakash Mirchandani[1]
University of Pittsburgh

This is joint work with Anant Balakrishnan[2] *of the University of Texas at Austin and Hari Natarajan of Penn State University.*

[1]Partially supported by NSF grant number DMI-0115434
[2]Partially supported by NSF grant number DMI-0115304

Outline

- Problem motivation
- Problem definition
- Cut formulation
- Basic flow formulations
- Model enhancements
- LP-based heuristic
- Computational results

Survivable Network Design: Problem Motivation

- Telecomm service disruptions are costly and catastrophic
  - "Just in case, Many Firms Work to Set Up Redundant Telecommunication Systems", *Wall Street Journal*, Dec. 20, 2001

- Telecomm companies must report to FCC outages that last > 30 minutes and affect > 30,000 customers

$\Rightarrow$ Need to ensure that service is uninterrupted, particularly for important customers, dense areas

- Logistics network disruption can cause significant financial loss
  - The West Coast ports shutdown last fall resulted in estimated losses of $1-2 billion/day (*New York Times,* Oct. 8, 2002*)*

Providing Uninterrupted Service

- Select robust network topology, containing multiple disjoint paths between critical nodes

SURVIVABLE NETWORK DESIGN problem

- Install spare capacity and hardware to instantaneously re-route flows when a link or node fails

NETWORK RESTORATION problem

Survivable Network Design: Principles

- Topology of network determines its ability to cope with disruptions and failures.
- Providing redundancy (alternate paths) for all possible flows can be prohibitively expensive.
- So, identify critical or *important* flows (or nodes), and ensure that these flows have alternate paths

Node importance levels

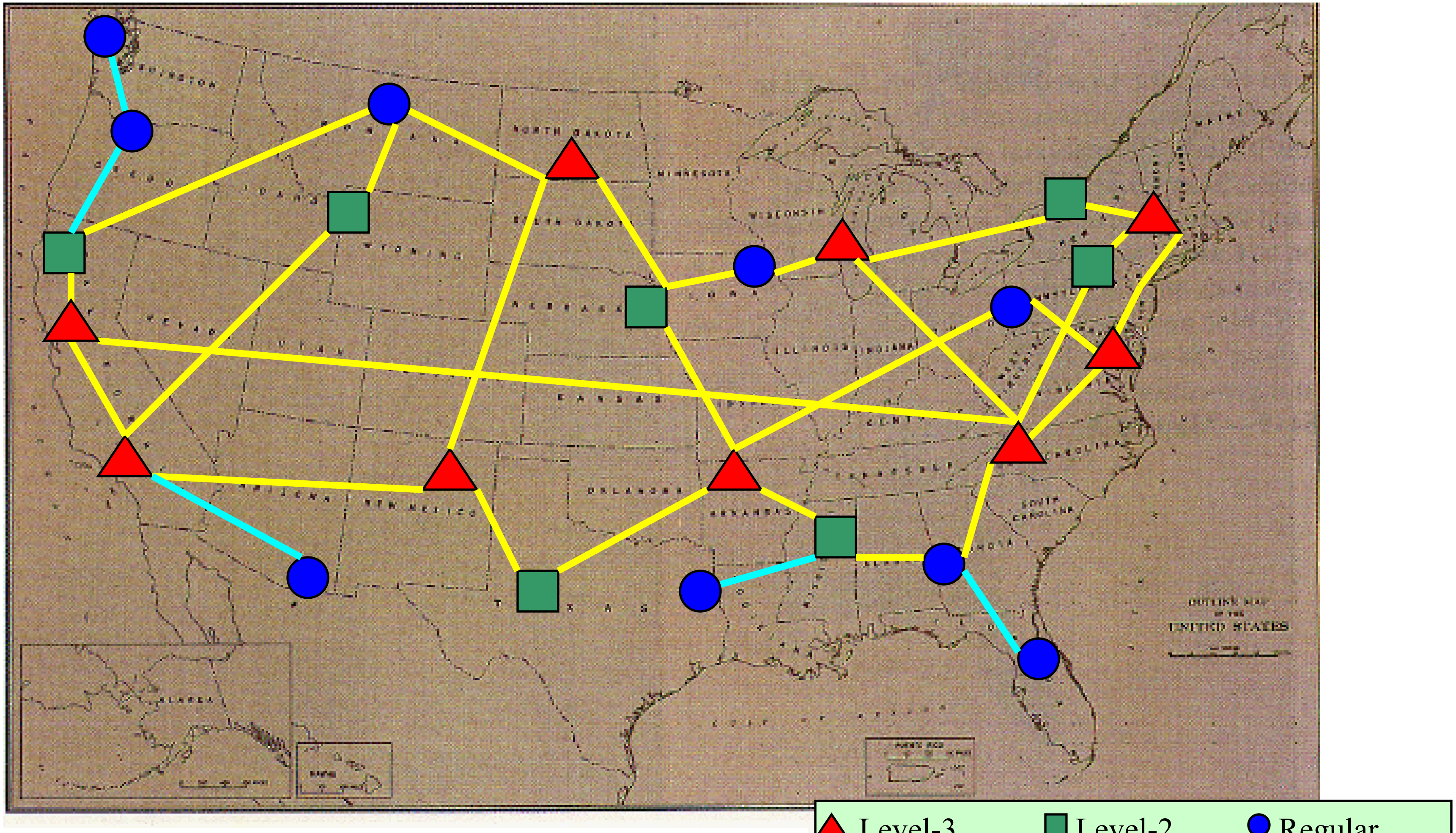
Critical nodes: require multiple edge-disjoint paths interconnecting them

Regular nodes: must be reachable, i.e., require one path

Steiner nodes: optional intermediate points

- For each node i, let $r_i \in \{0, 1, 2, \ldots\}$ be the *level* of node i
- Network must contain $min\ \{r_i, r_j\}$ edge-disjoint paths interconnecting node i to node j

Survivable Network Topology



US map courtesy http://info.er.usgs.gov/fact-sheets/maps-us/index.html

Survivable Network Design (SND) Problem

- **Given**:
  - location of nodes, and their importance levels (critical, regular, Steiner nodes)
  - possible network interconnections (edges)
  - fixed cost for each edge
- **Goal**:
  - Find minimum cost network configuration to meet all connectivity requirements with edge-disjoint paths

Notation

- $G{:}(V,E)$ given undirected network
- i, j nodes of network
  - Index the nodes in decreasing order of importance level. So, node 1 has highest importance level.
  - C, R, S = set of *C*ritical, *R*egular, and *S*teiner nodes
- r_{ij} minimum required number of edge-disjoint paths from node *i* to node *j*
- (i, j) edges of network
- c_{ij} cost of using edge (i, j) in the design

Classical Cutset Model

- $u_{ij} = \mathit{design}$ variable; =1 if edge (i, j) is selected, 0 otherwise
- Cutset $\{S, T\}$ = set of edges separating nodes of S from $T=V\backslash S$
- Need at least $q_{ST} = \max_{i\in S, j\in T}(r_{ij})$ edges across each cutset $\{S,T\}$

[*CUT*] $$Z^* = Z_{CUT} = \min \sum_{(i,j)\in E} c_{ij}u_{ij}$$

subject to:

$$\sum_{(i,j)\in\{S,T\}} u_{ij} \geq q_{ST} \quad \forall \text{ cutsets } \{S,T\}$$

$$u_{ij} \in \{0,1\} \qquad \forall\ (i, j) \in E.$$

Equivalent Flow Formulation: Full_Demand

- For every pair of nodes $k,\ l \in R \cup C$, define a *commodity <k,l>* with origin k, destination l, demand r_{kl}

[*FULL_DEMAND*]

$$Z_{FULL\_DEMAND} = \min \sum_{(i,j)\in E} c_{ij} u_{ij}$$

subject to:

$$\sum_j f_{ij}^{<k,l>} - \sum_j f_{ji}^{<k,l>} = \begin{cases} r_{kl} & \text{if } i = k \\ -r_{kl} & \text{if } i = l \\ 0 & \text{otherwise} \end{cases} \qquad \forall <k,l> \in K,$$

Forcing constraints

$$f_{ij}^{k} + f_{ji}^{k} \le u_{ij} \qquad \forall\ (i,j) \in E, k \in K,$$

$$f_{ij}^{k}, f_{ji}^{k} \ge 0 \qquad \forall\ (i,j) \in E, k \in K$$

$$u_{ij} \in \{0,1\} \qquad \forall\ (i,j) \in E.$$

Equivalent Flow Formulation: Tree_Demand

- $k \in K$ = index of *commodity* with origin 1, destination k

[*TREE_DEMAND*] $$Z_{TREE\_DEMAND} = \min \sum_{(i,j)\in E} c_{ij} u_{ij}$$

subject to:

$$\sum_j f_{ij}^k - \sum_j f_{ji}^k = \begin{cases} r_k & \text{if } i = 1 \\ -r_k & \text{if } i = k \\ 0 & \text{otherwise} \end{cases} \qquad \forall\ k \in K,$$

Forcing constraints $$f_{ij}^k + f_{ji}^k \le u_{ij} \qquad \forall\ (i,j) \in E, k \in K,$$

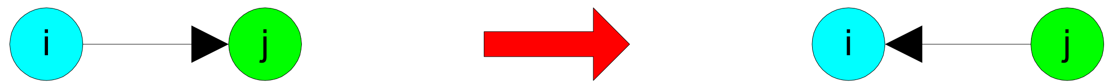
$$f_{ij}^k, f_{ji}^k \ge 0 \qquad \forall\ (i,j) \in E, k \in K$$

$$u_{ij} \in \{0,1\} \qquad \forall\ (i,j) \in E.$$

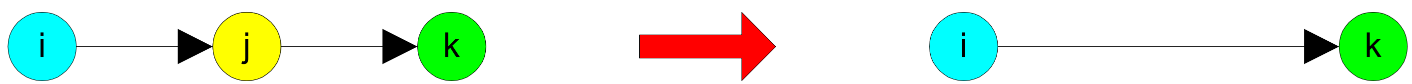
- O(n) commodities versus O(n^2) commodities for the Full-Demand case

Equivalence of FULL_DEMAND and TREE_DEMAND Formulations

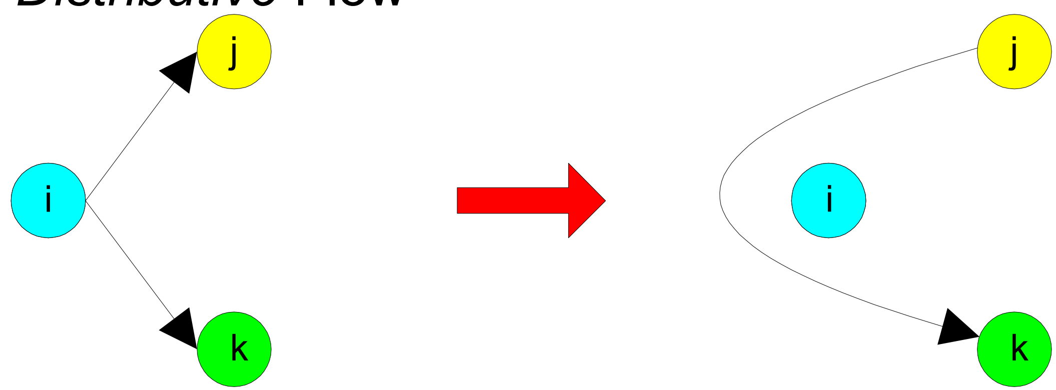
- **Proof based on the following elementary observations:**
- **Observation 1:** *Symmetric* Flow



- **Observation 2:** *Transitive* Flow



- **Observation 3:** *Distributive* Flow



Problem Complexity

- Generalizes several classical optimization problems
  - Traveling Salesman problem
  - Facility Location problem
  - Steiner Tree problem
- *NP-Hard* problem

Literature Review

- **Exact (polyhedral) methods**
    - Grötschel et al.,1992, 1995, 1997
    - Magnanti and Raghavan, 2002
    - Chopra and Rao, 1994, Goemans, 1994 (Steiner tree)
- **Approximate (heuristic) approaches**
  - Based on structural properties
    - Monma and Shallcross, 1989
    - Goemans and Bertsimas, 1993
    - Balakrishnan, Magnanti and Mirchandani, 2002
  - Primal-dual
    - Goemans and Williamson, 1995
    - Williamson, Goemans, Mihail, Vazirani, 1995
    - Goemans, Goldberg, Plotkin, Shmoys, Tardos, Williamson, 1994
  - Linear programming-based
    - Jain, 2001

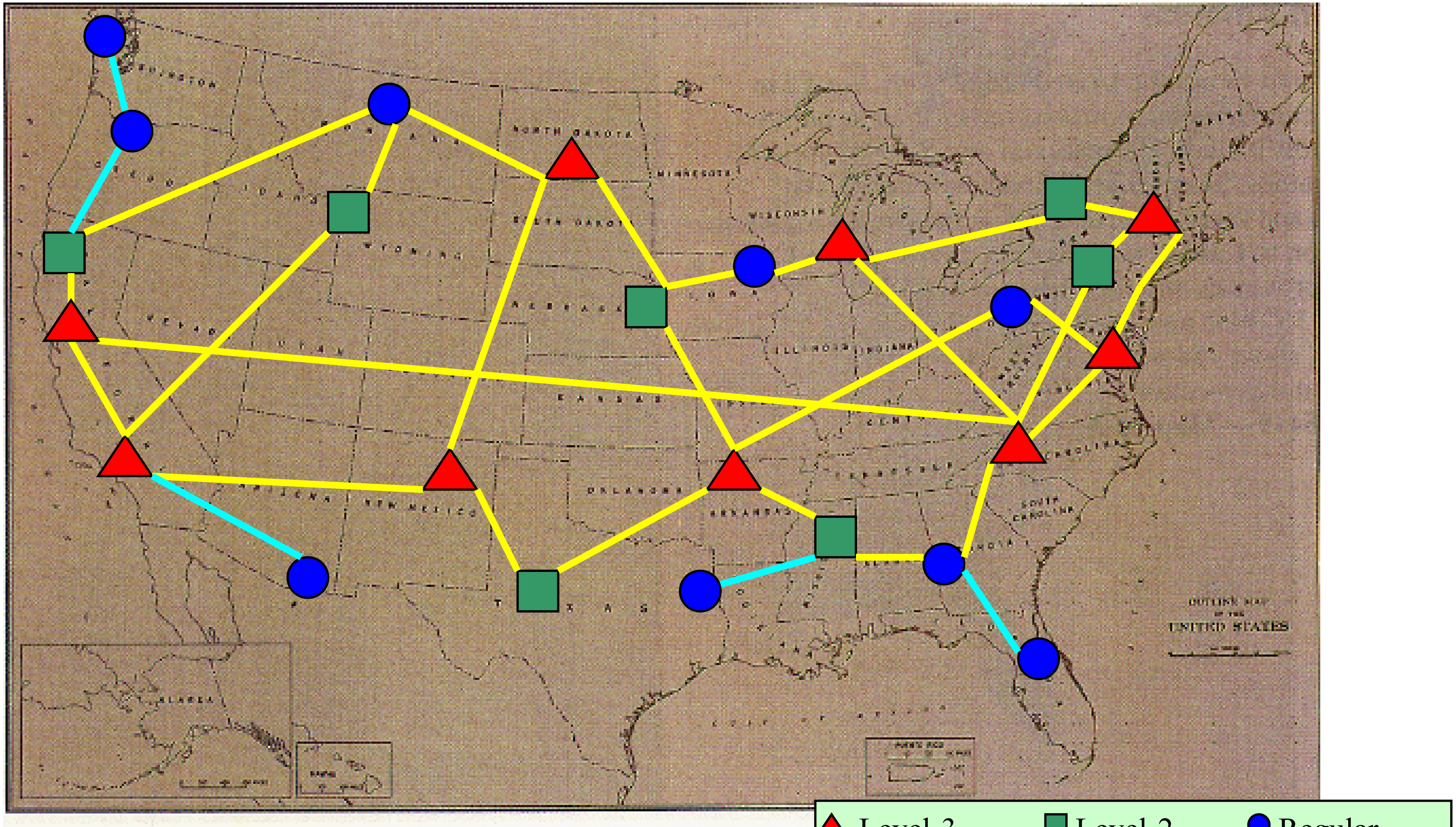
Solution Approach

- Generate tight lower bounds
  - Identify and add valid inequalities to increase LP relaxation value
- Generate good heuristic solutions
  - LP-rounding procedure
- If gap between upper and lower bound is small, stop. Else, possibly use branch-and-cut procedure

Strengthening the SND Model

- "Upgrade" the node importance levels, if possible
  - Upgrade the *Regular* nodes in the backbone (multi-connected) network to Critical nodes
  - Upgrade the *Steiner* nodes in the access (backbone) network to Regular (Critical) nodes
- Strengthen the *forcing* constraints
- Add *cardinality* and (conditional) *degree* constraints

Survivable Network Topology



US map courtesy http://info.er.usgs.gov/fact-sheets/maps-us/index.html

Extended Flow Model

t_k = Node-level upgrade *variable*

[*EXT_FLOW*] $Z_{EXT\_FLOW} = \min \sum_{(i,j)\in E} c_{ij} u_{ij}$

subject to:

$$\sum_j f_{ij}^k - \sum_j f_{ji}^k = \begin{cases} t_k & \text{if } i = 1 \\ -t_k & \text{if } i = k \\ 0 & \text{otherwise} \end{cases} \quad \forall\ \mathrm{k} \in V \setminus \{1\},$$

$$f_{ij}^k + f_{ji}^k \le u_{ij} \qquad \forall\ (i,j) \in E, l \in V \setminus \{1\},$$

$$t_i \ge r_i \qquad \forall\ i \in V \setminus \{1\}$$

$$f_{ij}^k, f_{ji}^k \ge 0 \qquad \forall\ (i,j) \in E, l \in V \setminus \{1\},$$

$$u_{ij} = 0 \text{ or } 1 \qquad \forall\ (i,j) \in E$$

+ Additional node upgrade constraints on t_i

Node Upgrade constraints

- **U1** "Criticalize" Regular nodes based on Critical flows

$$t_r \geq (f_{rj}^c + f_{jr}^c) + 1 \qquad \forall r \in R, (r, j) \in E, c \in C$$

- **U2** "Regularize" Steiner nodes

$$t_s^{'} \geq u_{sj} \qquad \forall s \in S, (s, j) \in E$$

- **U3** "Criticalize" Steiner nodes based on Critical flows

$$t_s \geq (f_{sj}^c + f_{js}^c) + t_s^{'} \qquad \forall s \in S, (s, j) \in E, c \in C$$

- **U4** "Criticalize" Regular/Steiner nodes based on criticalized Regular/Steiner flows (bootstrap)

$$t_l \geq (f_{lj}^k + f_{jl}^k) + t_k - 1 \qquad \forall l, k \in S \cup R, l \neq k, (l, j) \in E$$

Node Upgrade (contd)

- **U5** “Criticalize” Regular nodes with two critical neighbors

$$t_r \geq u_{rc} + u_{rc} \forall r \in R; c, c' \in C, c \neq c'; (r,c), (r,c') \in E$$

Bi-directional Forcing constraints

- **BF1** Regular-Regular Forcing constraints

$$f_{ij}^{r} + f_{ji}^{r'} \le u_{ij} + \frac{t_r + t_{r'}}{2} - 1 \qquad \forall r, r' \in R, r \ne r'; (i,j) \in E$$

$$f_{ji}^{r} + f_{ij}^{r'} \le u_{ij} + \frac{t_r + t_{r'}}{2} - 1 \qquad \forall r, r' \in R, r \ne r'; (i,j) \in E$$

- **BF2** Regular-Critical Forcing constraints

$$f_{ij}^{r} + f_{ji}^{r} + f_{ij}^{c} + f_{ji}^{c} \le u_{ij} + \frac{t_r}{2} \qquad \forall r \in R, c \in C, (i,j) \in E$$

- **BF3** Regular-incident Forcing constraints

$$f_{rj}^{k} + f_{jr}^{k'} \le u_{rj} + t_r - 1 \quad \forall r \in R; k, k' \in R \cup S, k \ne k', r; (r,j) \in E$$

$$f_{jr}^{k} + f_{rj}^{k'} \le u_{rj} + t_r - 1 \quad \forall r \in R; k, k' \in R \cup S, k' \ne k, r; (r,j) \in E$$

Design constraints

- **D1** Regular node degree constraints

$$\sum_{j} u_{rj} \geq \sum_{i} f_{ir}^{k} + 1 \qquad \forall r \in R, k \in V \setminus \{1\}$$

- **D2** Design cardinality constraints

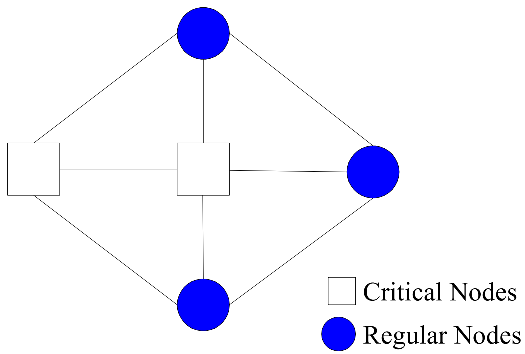
$$\sum_{(i,j)\in E} u_{ij} \geq |C| + |R| + \sum_{s\in S} t_s^{'}$$

Hierarchy of SND (Flow) Models

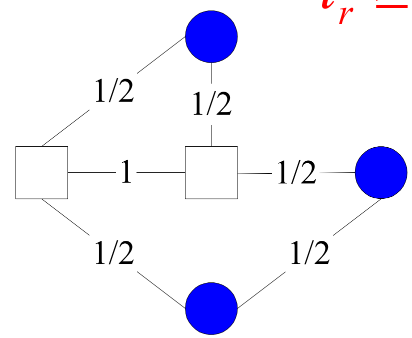
- Base Model: Tree_Flow
- Node Upgrade Model: Base model + criticalize regular nodes
- Strong Model: Other constraints
  - Extended Regular node upgrades
  - Steiner node upgrades
  - Strong bi-directional constraints
  - Degree and cardinality constraints

Comparison of Flow Formulation solutions

$$t_r \geq (f_{rj}^c + f_{jr}^c) + 1$$

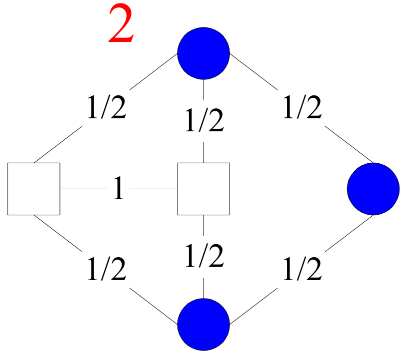


Problem Instance. All edge costs equal 1.

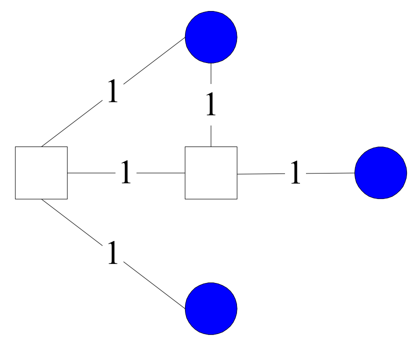


LP Solution for **Base** Formulation
Cost = 3.5

$$f_{ij}^r + f_{ji}^{r'} \leq u_{ij} + \frac{t_r + t_{r'}}{2} - 1$$



LP Solution for **Node Upgrade** Formulation
Cost =4



LP Solution for **Strong** Formulation
Cost = 5. Also, the **Optimal** Solution

Benefits of Strong SND Formulation

- The new node-upgrade models are **stronger** than the traditional cutset formulation.
  - provides better LP-based lower bounds
  - for some problem instances, strong LP gives optimal integer solution
  - improves computational performance of branch-and-bound algorithm
  - provides better, quicker heuristic solutions
- Goal: Evaluate computational effectiveness of strong formulation through empirical testing

Determining Upper Bound: Iterative LP-rounding heuristic

Step 1: Solve LP relaxation of SND model

Step 2: Round-up a fractional edge value u_{ij} to 1

- Selecting the fractional edge
  - max u_{ij}
  - max “utilization”
  - min incremental rounding cost per unit of flow

Step 3: Re-solve LP relaxation with fixed edges

If solution is integer-valued, Stop

Else, repeat Step 2

Computational Testing

- Objectives
  - Develop and implement optimization-based methodology for SND problem
  - Compare LP relaxations of Base and Enhanced SND models
  - Evaluate the effectiveness of LP relaxation and heuristics
- Computational Platform
  - Dual Processor 933 MHz, 2 GB Ram; Windows 2000
  - LP solver: CPLEX Version 7.5
- Test problems
  - randomly generated, Euclidean
  - parameters to vary network size and topology, distribution of node levels
  - 3 instances per problem type

Computational Results: Summary

Problem Size	S-R-C node proportions	Percentage Gap (%)
# of Nodes: 20 # of Arcs: 80	0-25-75	0.7%
	0-50-50	0.1%
	0-75-25	0.0%
	20-40-40	0.0%
	40-30-30	0.0%
	Average	***0.16%***
	# Strong LP closes gap	12 of 15
# of Nodes: 30 # of Arcs: 120	0-25-75	0.0%
	0-50-50	0.9%
	0-75-25	0.6%
	20-40-40	0.3%
	40-30-30	0.4%
	Average	***0.44%***
	# Strong LP closes gap	9 of 15
# of Nodes: 40 # of Arcs: 160	0-25-75	0.6%
	0-50-50	0.3%
	0-75-25	0.6%
	20-40-40	2.5%
	40-30-30	0.2%
	Average	***0.82%***
	# Strong LP closes gap	3 of 15

Computational Results: LP effectiveness

Problem Size	S-R-C node proportions	Integrality Gap (%)			% Gap Reduction
		Base	Node Upgrade	Strong	
# of Nodes: 20 # of Arcs: 80	0-25-75	3.1%	0.8%	**0.7%**	*78.4%*
	0-50-50	13.0%	4.8%	**0.1%**	*99.0%*
	0-75-25	26.4%	16.2%	**0.0%**	*99.8%*
	20-40-40	10.9%	3.4%	**0.0%**	*99.7%*
	40-30-30	12.1%	5.8%	**0.0%**	*100.0%*
# of Nodes: 30 # of Arcs: 120	0-25-75	4.5%	1.2%	**0.0%**	*100.0%*
	0-50-50	17.1%	9.2%	**0.9%**	*93.9%*
	0-75-25	25.6%	15.4%	**0.6%**	*97.0%*
	20-40-40	14.6%	7.6%	**0.3%**	*98.2%*
	40-30-30	15.8%	7.8%	**0.4%**	*97.1%*
# of Nodes: 40 # of Arcs: 160	0-25-75	4.2%	0.9%	**0.6%**	*83.5%*
	0-50-50	13.3%	2.2%	**0.3%**	*97.2%*
	0-75-25	29.4%	17.6%	**0.6%**	*97.4%*
	20-40-40	14.1%	7.3%	**2.5%**	*81.0%*
	40-30-30	12.3%	6.2%	**0.2%**	*97.6%*
Average	0-25-75	4.0%	1.0%	**0.4%**	*87.3%*
	0-50-50	14.5%	5.4%	**0.4%**	*96.7%*
	0-75-25	27.1%	16.4%	**0.4%**	*96.0%*
	20-40-40	13.2%	6.1%	**0.9%**	*93.0%*
	40-30-30	13.4%	6.6%	**0.2%**	*98.2%*

Computational Results (contd)

Problem Size	Node Proportion by Level		Percentage Gap (%)
# of Nodes: 40 **# of Arcs: 160**	0-1-2-3 Problem	25-25-25-25	2.1%
# of Nodes: 100 **# of Arcs: 400**	0-1-2 Problem	50-0-50	0.9%

Using Base model, CPLEX required over 4 hours to solve 40 node problem instance

Heuristic performance

Problem Size	S-R-C node proportions	Heuristic Performance	
		Max u	Cost/Unit Flow
# of Nodes: 20 # of Arcs: 80	0-25-75	2 of 3	2 of 3
	0-50-50	2 of 3	3 of 3
	0-75-25	3 of 3	3 of 3
	20-40-40	3 of 3	3 of 3
	40-30-30	3 of 3	3 of 3
	Total	**13 of 15**	**14 of 15**
# of Nodes: 30 # of Arcs: 120	0-25-75	3 of 3	3 of 3
	0-50-50	3 of 3	2 of 3
	0-75-25	3 of 3	3 of 3
	20-40-40	2 of 3	3 of 3
	40-30-30	2 of 3	3 of 3
	Total	**13 of 15**	**14 of 15**
# of Nodes: 40 # of Arcs: 160	0-25-75	2 of 3	2 of 3
	0-50-50	2 of 3	2 of 3
	0-75-25	2 of 3	2 of 3
	20-40-40	2 of 3	2 of 3
	40-30-30	2 of 3	3 of 3
	Total	**10 of 15**	**11 of 15**

Conclusions

- Strong problem formulations are very effective for solving difficult SND problems
- LP-based heuristic, based on strong formulation, performs well
- *Min incremental cost per unit of flow* method is superior to *max* u_{ij} method
- Further work
  - More testing for higher connectivity problems
  - More comparison with IP solution times
  - Polyhedral results